REMARKS

In the final Office Action, the Examiner rejected Claims 1, 2, 4 – 11 and 19 – 22 under 35 U.S.C. §103(a) as being unpatentable over Windows™ Explorer and Ku et al. in view of Pajak et al. Claims 3 and 12 – 18 were rejected under 35 U.S.C. §103(a) as being unpatentable over Windows™ Explorer, Ku et al. and Pajak et al. in view of Rich et al.

Applicants have amended Claim 1 and canceled Claims 2 – 22. Claims 23 – 39 have been added for consideration. Support for the new claims (e.g., Claims 1, 28 and 34) can be found on page 8, line 21 to page 10, line 19 as well as and Figs. 3B – 3E (see elements 320 – 350). Thus, no new matter has been to the Application.

By this amendment Claims 1, 23 – 39 are pending. For the reasons stated more fully below, Applicants submit that the claims are allowable over the applied references. Hence, reconsideration, allowance and passage to issue are respectfully requested.

As stated in the SPECIFICATION, in a typical distributed data processing system, there are many situations where a user may need to display and interact with both local and remote data objects. Local and remote data objects may include, for example, files and folders located locally and remotely. Typical actions performed on these local and remote data objects may include, for example, editing, deleting, copying, moving, renaming, and in the case of program files and executable files, compiling and running.

Known data object viewers typically display either all local or all remote data objects in a single view. When interacting with remote data objects presented in such a view, two approaches have been commonly used. In a first approach, an action on a remote data object is invoked directly on the remote system, for example by programming a customized remote action. In a second approach, a remote data object is "pulled" or downloaded to a local system, acted upon locally, and then "pushed" or uploaded back to the remote system.
CA920030063US1

For example, there are Windows-based code editors, which allow for local editing of remote files. Typically, this is done by downloading a temporary copy of the remote files to a local system, allowing the files to be edited locally, and then subsequently pushing the edited files back to the remote system. Thus, there is a need to provide a more flexible approach to displaying and interacting with local and remote data objects.

The present invention provides such a flexible approach. In accordance with the teachings of the invention, both local data objects and remote data objects may be displayed in a viewer. In the case where a particular data object is both a local data object and a remote data object, the data object may be displayed as a hybrid data object representing both the local data object and the remote data object. When a user selects a displayed hybrid data object on which to perform an action, the actual data object on which the action is to be performed will be requested. Once the user indicates on which data object the action is to be performed, it will be done.

The act of requesting on which actual data object to perform the action is rather important since action can only be performed on a data object as permitted by the system on which the data object is located.

The invention is set forth in claims of varying scopes of which Claim 1 is illustrative.

> 1. A method of interacting with local and remote data objects in a distributed data processing system, comprising:
>
> determining whether one remote system in the distributed data processing system and a local system have a data object in common;
>
> ***displaying*** on the local system, if it is determined that the remote system in the distributed data processing system and the local system have a data object in common, ***the data object as a hybrid data object, the hybrid data object representing both the data object on the local system and the remote system***;

> enabling a user on the local system to perform an action on the hybrid data object by first selecting the hybrid data object;
>
> ***prompting the user, when the user selects the hybrid data object, to indicate whether the action is to be performed on the local data object, the remote data object or both the local data object and remote data object***; and
>
> performing the action as indicated by the user. (Emphasis added.)

Applicants submit that the claims are allowable over the applied references because none of the references (i.e., Windows™ Explorer, Ku et al., Pajak et al. and Rich et al.) teaches, shows or so much as suggests the steps of ***displaying a data object as a hybrid data object that represents both a local data object and a remote data object and prompting the user, when the user selects the hybrid data object, to indicate whether the action is to be performed on the local data object, the remote data object or both the local data object and remote data object*** as claimed.

For example, it is well known that Windows™ Explorer does not teach such steps.

Ku et al., on the other hand, teach a search facility for local and remote interface repositories. According to the teachings of Ku et al., one or more CORBA Interface Repositories may be searched for program objects based upon a set of user-specified search criteria. A user interface screen or frame is provided with two panes, the first of which allows a user to specify a variety of criteria for which to search one or more of the Interface Repositories using form fields, radio buttons, and/or drop-down lists. The second display pane provides a textual listing of the objects found, their location or locations, their revision dates. Searches may be stored for later review or transmission to other development team members.

Pajak et al purport to teach a multi-user collaborative system. According to the purported teachings of Pajak et al., different users at different workstations

PAGE 12/14 * RCVD AT 5/1/2006 5:00:53 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/6 * DNIS:2738300 * CSID:512 306 0240 * DURATION (mm-ss):03-50

connected to a common link may access a shared structured data object (i.e., a file). However, editing or modification cannot be performed by a user until the shared structured data object has been locked up to prevent multiple users from editing or modifying the shared data object at the same time. Visual indication as to the (locked) state of the shared data object and other information relative to the locking user and the time of lock is updated and displayed in the shared structured data object representation present at user workstations when a user invokes a user operation on the shared structured data object or its contents. In this manner, the updating of the representation is completely decentralized and client-based so that it is not necessary to monitor the number and currency of shared structured data objects existing throughout the network but rather, updating of the representation of object content, as well as any modified data content of structured data objects, is accomplished upon individual user initial invoking of a structured data object operation.

Rich et al., teach a transaction-scoped replication for distributed object systems. According to the teachings of Rich et al. a remote object is replicated to a node of a distributed system from which it is accessed. The scope of the replication is a transaction. Thereafter, method invocations on the object occur locally, avoiding the performance overhead of frequent round trips to the remote persistent object store. Changes made to a replicated object by a transaction are represented using a tree structure that is internally managed by the application. When an application or application user has made modifications to a replicated object and requests to commit the modifications, a determination is first made as to whether committing the modifications will result in an unacceptable data conflict. If no unacceptable data conflict will occur, and after resolution of those conflicts that can be resolved, the modifications are committed. Nested transactions are supported, where each child transaction may commit or roll back independently.

Since none of the applied references teaches the emboldened/italicized limitations in the above-reproduced Claim 1, Applicants submit that Claim 1, as

PAGE 13/14 * RCVD AT 5/1/2006 5:00:53 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/6 * DNIS:2738300 * CSID:512 306 0240 * DURATION (mm-ss):03-50

well as its dependent claims, should be allowable. Independent Claims 28 and 34, which all incorporate the above-emboldened-italicized limitations in the above-reproduced claim 1, together with their dependent claims, should also be allowable. Hence, Applicants once more respectfully request reconsideration, allowance and passage to issue of the claims in the application.

Respectfully Submitted

By:

Volel Emile
Attorney for Applicants
Registration No. 39,969
(512) 306-7969

PAGE 14/14 * RCVD AT 5/1/2006 5:00:53 PM [Eastern Daylight Time] * SVR:USPTO-EFXRF-5/6 * DNIS:2738300 * CSID:512 306 0240 * DURATION (mm-ss):03-50